Welcome to Your Team's Cloud Environment

This guide will help you access the cloud resources set up for your class with Katerina Doka. Follow these steps to set up the necessary tools, securely download your private key, and connect to your team's virtual machines (VMs).

Understanding Your Team's Setup

Each team has a dedicated environment that includes:

Bastion Host (Entry Point)

- This is your gateway into the cloud environment.
- It's the only machine with a public IP address, allowing you to connect from your personal computer.
- The Public IP of the bastion host is listed in `team_{Number}_ips.csv`.

Private VMs (Team Machines)

- Each team has four private VMs for your work.
- These machines don't have public IPs and can only be accessed after logging into the bastion host.
- Their private IPs are listed in `team_{Number}_ips.csv` under the `PrivateIps` column (comma-separated).
- The private VMs are **t3.xlarge EC2 instances**, which means:
  - 4 vCPUs (virtual processors)
  - 16GB RAM
  - 100GB of storage

Private Key Storage

- Your private SSH keys, which allow you to connect to your machines, are stored securely in an AWS S3 bucket.
- Only your team has access to these keys, ensuring security and isolation.

Team Isolation

- Each team operates in its own private space, ensuring security and preventing interference between teams.
- Access is managed through AWS Identity and Access Management (IAM) and SSH keys.

Step 1: Setting Up AWS Command Line Interface (CLI)

Why This Is Important

The AWS CLI lets you interact with AWS services from your terminal, making it easier to manage your resources.

Install AWS CLI

macOS: Use Homebrew:

```
brew install awscli
```

Windows: Download and install AWS CLI:

1. Visit the AWS CLI download page.
2. Download and run the Windows installer.
3. Follow the installation instructions.
4. Verify installation by running:

   ```
   aws --version
   ```

Linux: Use your package manager:

```
sudo apt-get install awscli
```

Configure AWS CLI

After installation, configure it with your credentials:

```
aws configure
```

You'll be prompted to enter:

- **AWS Access Key ID:** Found in `team_{Number}_credentials.csv`.
- **AWS Secret Access Key:** Also in `team_{Number}_credentials.csv`.
- **Default region name:** Use `eu-central-1`.
- **Default output format:** Use `json`.

Step 2: Automated Setup Scripts

Automated Configuration Scripts

To simplify the setup process, your folder contains two scripts that automate the steps described below:

- **For macOS/Linux:** `team_setup.sh`
- **For Windows:** `team_setup.ps1`

These scripts will:

- Read the necessary details from `team_{Number}_ips.csv` and `team_{Number}_credentials.csv`.
- Automatically configure the AWS CLI with the correct credentials.
- Download and properly set permissions for the SSH private key.
- Set up SSH configurations for easier logins.
- Create convenient helper scripts for connecting to your machines.

Running the Scripts

Ensure the script is executable before running it:

For macOS/Linux:

```
chmod +x team_setup.sh
./team_setup.sh
```

For Windows (PowerShell):

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
./team_setup.ps1
```

Successful Setup Outcome

After running the setup script successfully, three new helper scripts will be created in your team folder:

- **connect_bastion.sh** - Runs the command to connect you to the bastion host:

  ```
  ./connect_bastion.sh
  ```

- **connect_vm.sh <vm_number>** - Connects you to a specific VM by specifying the number:

  ```
  ./connect_vm.sh 1  # Connects to VM 1
  ```

- **connect_all.sh** - Opens connections to all VMs in separate terminal windows:

  ```
  ./connect_all.sh
  ```

Additionally, SSH aliases have been set up for easier access:

- `ssh team-bastion` - Connects to the bastion host.
- `ssh team-vm1` - Connects to VM 1 through the bastion host.

- `ssh team-vm2` - Connects to VM 2 through the bastion host.

Manual Setup (Alternative Method)

Step 3: Downloading Your Private SSH Key Manually

For macOS/Linux:

```
aws s3 cp s3://distributed-systems-course-team-keys/{TEAM_NUMBER}/private_key.pem ~/.ssh/team_key.pem
chmod 600 ~/.ssh/team_key.pem
```

For Windows (PowerShell):

```
aws s3 cp s3://distributed-systems-course-team-keys/{TEAM_NUMBER}/private_key.pem C:\Users\$env:USERNAME\.ssh\team_key.pem
icacls C:\Users\$env:USERNAME\.ssh\team_key.pem /inheritance:r
icacls C:\Users\$env:USERNAME\.ssh\team_key.pem /grant:r $env:USERNAME:F
```

Step 4: Manually Setting Up SSH Configurations

To enable shortcut commands for easier access:

For macOS/Linux:

Edit your SSH config file:

```
touch ~/.ssh/config
nano ~/.ssh/config
```

Add the following lines (replace `<PublicIP>` and `<PrivateIP>`):

```
Host team-bastion
  HostName <PublicIP>
  User ubuntu
  IdentityFile ~/.ssh/team_key.pem

Host team-vm1
  HostName <PrivateIP_1>
  User ubuntu
  IdentityFile ~/.ssh/team_key.pem
  ProxyJump team-bastion
```

```
Host team-vm2
  HostName <PrivateIP_2>
  User ubuntu
  IdentityFile ~/.ssh/team_key.pem
  ProxyJump team-bastion
```

Save and exit. Now, you can connect using:

```
ssh team-bastion
ssh team-vm1
ssh team-vm2
```

For Windows (PowerShell/Git Bash):

Modify `C:\Users\$env:USERNAME\.ssh\config` similarly to the macOS/Linux setup above.

Troubleshooting

1. **AWS CLI Access Denied:** Check your credentials in `team_{Number}_credentials.csv`.
2. **SSH Permission Denied:** Ensure key permissions are set (`chmod 600 ~/.ssh/team_key.pem`).
3. **PowerShell Script Blocked:** Run `Set-ExecutionPolicy RemoteSigned` before executing scripts.
4. **Connection Timeout:** Ensure you are using the correct IPs from `team_{Number}_ips.csv`.

By following these steps, you'll have a secure and efficient setup to work within your team's cloud environment.